WHAT IS CLAIMED IS:

1. A preprocessor, wherein a first circuit description file containing a first hardware description language and a second hardware description language is processed on the basis of a preprocessor control file, and at least a portion described by the first hardware description language in the first circuit description file is converted into the second hardware description language to create and output a second circuit description file.

2. A preprocessor according to claim 1, wherein a logic synthesis control script file for a gated clock circuit is further created and output on the basis of the preprocessor control file and the first circuit description file.

3. A preprocessor according to claim 1, wherein the second hardware description language includes one of Verilog-HDL and VHDL.

4. A preprocessor according to claim 1, wherein the processing of the first circuit description file is processing of extracting the first hardware description language from the first circuit description file, converting the first hardware description language into the second hardware description language, and outputting the second circuit description file without converting a portion described by the second hardware description language in the first circuit description

file.

5. A preprocessor according to claim 1, wherein the first circuit description file includes a description about a flip-flop, which is described by the first hardware description language, and information corresponding to a circuit obtained by forming a gated clock of the description about the flip-flop is converted into the second hardware description language and output.

6. A preprocessor according to claim 5, wherein the description about the flip-flop, which is made by the first hardware description language, is made without specifying a reset scheme of the flip-flop, and whether the flip-flop should use a synchronous reset scheme or an asynchronous reset scheme is designated in converting at least the portion described by the first hardware description language in the first circuit description file into the second hardware description language.

7. A preprocessor according to claim 5, wherein in the description about the flip-flop, one cluster number is assigned to one flip-flop description.

8. A preprocessor according to claim 5, wherein the second circuit description file is obtained by converting descriptions of flip-flops indicated by a plurality of cluster numbers into a description of a flip-flop to be driven by one gated clock signal.

9. An integrated circuit design system comprising:

a preprocessor which processes a first circuit description file containing a description of a flip-flop described by a first hardware description language on the basis of a preprocessor control file, creates a second circuit description file by converting at least the description of the flip-flop into a second hardware description language, and creates a logic synthesis control script file for a gated clock circuit; and

a logic synthesis tool which subjects the second circuit description file to logical synthesizes using logic synthesis control script file created by the preprocessor and converts the files into a circuit description file using a cell as a basic unit of a circuit to create a netlist and determine a layout of cells and wirings in an integrated circuit on the basis of the netlist.

10. A system according to claim 9, wherein the netlist is created by logically synthesizing, in addition to the second circuit description file and logic synthesis control script file, a logic synthesis control script file for control other than a gated clock.

11. A system according to claim 9, wherein information for gated clock formation of the flip-flop

is created by logic synthesis by a logic synthesis tool.

12. A system according to claim 9, wherein the description about the flip-flop, which is made by the first hardware description language, is made without specifying a reset scheme of the flip-flop, and whether the flip-flop should use a synchronous reset scheme or an asynchronous reset scheme is designated in converting at least the portion described by the first hardware description language in the first circuit description file into the second hardware description language.

13. A system according to claim 9, wherein in the description about the flip-flop, one cluster number is assigned to one flip-flop description.

14. A system according to claim 9, wherein the second circuit description file is obtained by converting descriptions of flip-flops indicated by a plurality of cluster numbers into a description of a flip-flop to be driven by one gated clock signal.

15. An integrated circuit design method comprising:

inputting, to a preprocessor, a circuit description file containing a first hardware description language and a second hardware description language and a preprocessor control file which controls operation of the preprocessor and converting at least

a portion described by the first hardware description language in the circuit description file into the second hardware description language;

logically synthesizing a circuit description file output from the preprocessor using a logic synthesis control script file for a gated clock circuit and a logic synthesis control script file for circuits other than the gated clock circuit by a logic synthesis tool to convert the files into a circuit description file using a cell as a basic unit of a circuit and create a netlist; and

determining the layout of the cells and the wirings on the basis of the netlist to design a circuit of a chip.

16. A method according to claim 15, wherein the circuit description file includes a description about a flip-flop, which is made by the first hardware description language, and a synchronous/asynchronous designation and cluster combination of a gated clock are changed by the preprocessor.

17. A method according to claim 16, wherein the preprocessor control file contains at least one of pieces of information representing a name of a reset signal, a name of a clock signal, a flip-flop reset scheme, whether a gated clock is to be formed, if the gated clock is to be formed, whether a description of clock gating is to be created by the preprocessor,

whether the description is to be output in a description format that can be automatically recognized by an automatic gated clock formation function of a logic synthesis program, and information about clusters to be regarded as one.

18. A method according to claim 15, wherein converting at least the portion described by the first hardware description language in the circuit description file into the second hardware description language comprises

loading and interpreting the preprocessor control file and storing the interpreted information in the preprocessor,

determining whether the information stored in the preprocessor is related to an extended description, and if the information is related to the extended description, analyzing the extended description,

determining a tag when it is determined that the information is not related to the extended description,

determining a cluster combination designation when it is determined that the tag is a predetermined tag,

designating whether the flip-flop should use synchronous reset or asynchronous reset and creating a circuit description corresponding to the flip-flop of synchronous reset or a circuit description corresponding to the flip-flop of asynchronous reset, and

creating information necessary for cluster

combination when the cluster combination designation is present.

19. A method according to claim 18, wherein analyzing the extended description comprises

determining whether the extended description is an extended description about the flip-flop,

storing information in each variable of a structure flip-flop when it is determined in the determination that the extended description is the extended description about the flip-flop, and

adding a tag to a pointer to an entity of the structure flip-flop and storing the tag in a temporary buffer.

20. A method according to claim 18, wherein determining the tag comprises

acquiring the tag and stored information from the temporary buffer, and

determining whether the tag is the predetermined tag, and when the tag is not the predetermined tag, directly outputting the stored information to an output file.